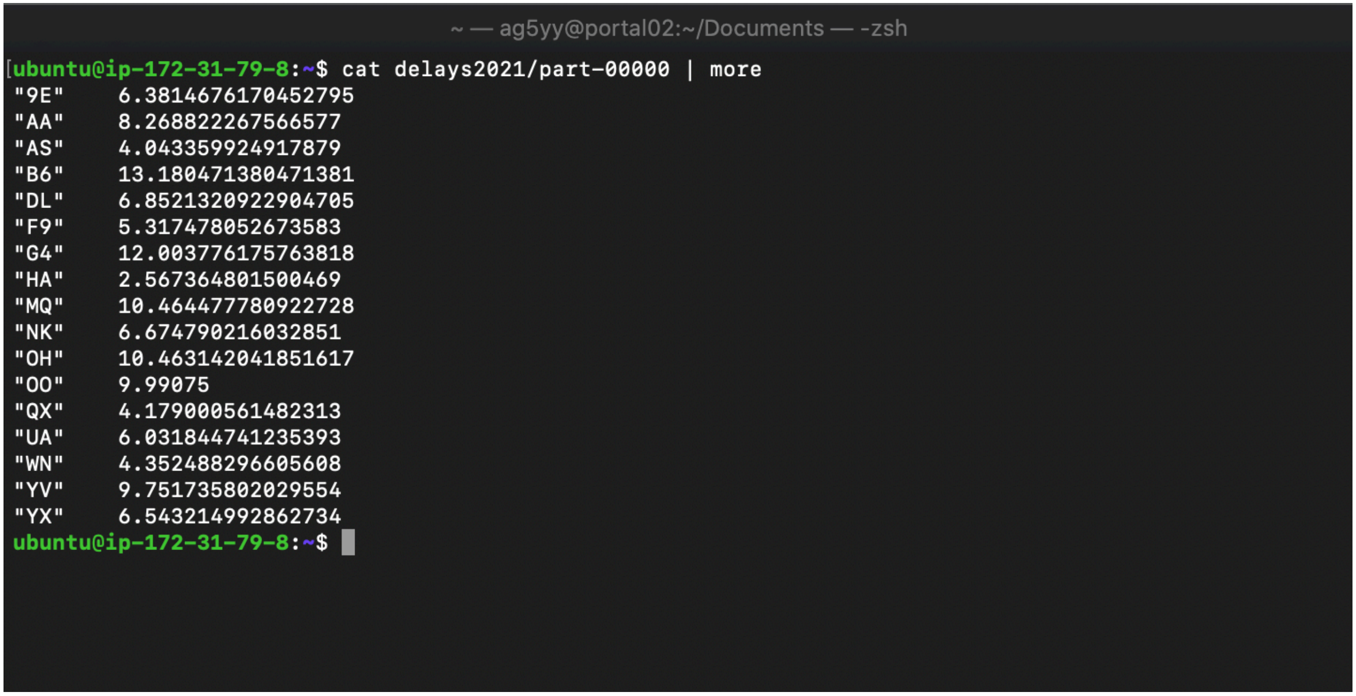
Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## Task 1:

### JAN 2021

***Best on-time performance:*** *HA, AS, QX, WN, F9*

***Worst on-time performance:*** *B6, G4, MQ, OH , OO*

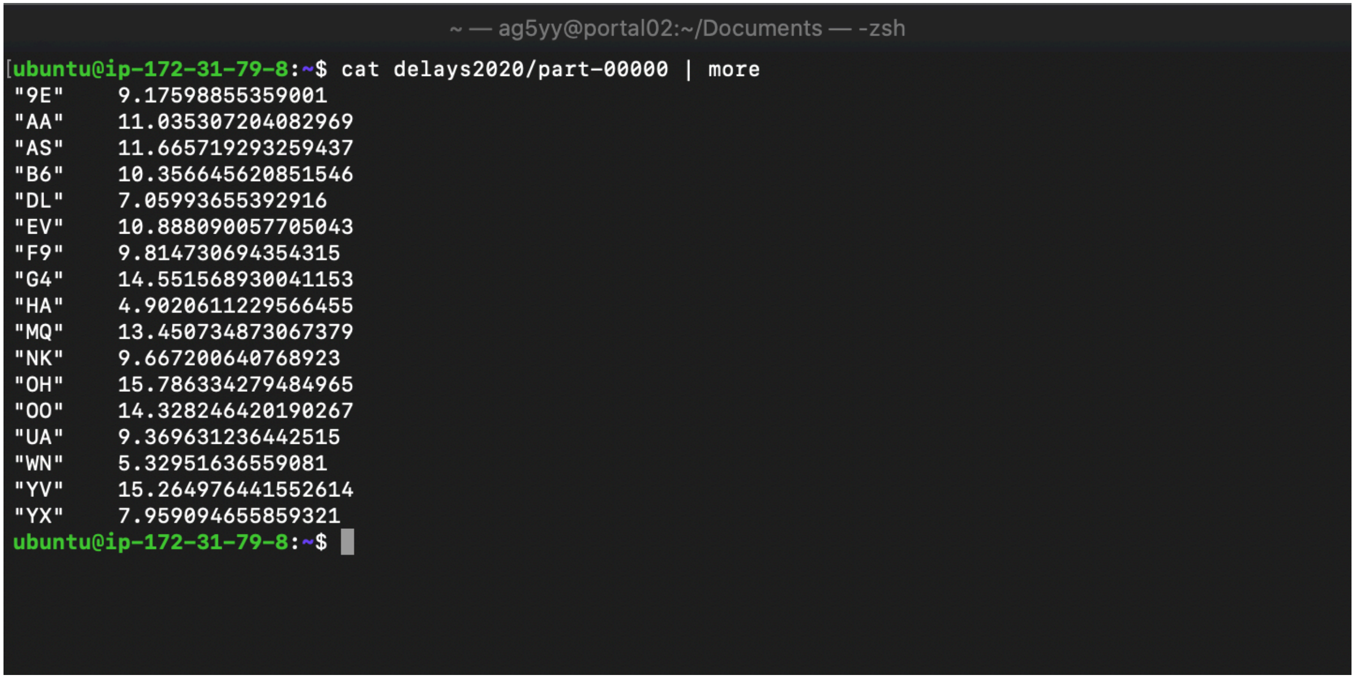


```
[ubuntu@ip-172-31-79-8:~$ cat delays2021/part-00000 | more
 "9E"    6.3814676170452795
 "AA"    8.268822267566577
 "AS"    4.043359924917879
 "B6"    13.180471380471381
 "DL"    6.8521320922904705
 "F9"    5.317478052673583
 "G4"    12.003776175763818
 "HA"    2.567364801500469
 "MQ"    10.464477780922728
 "NK"    6.674790216032851
 "OH"    10.463142041851617
 "OO"    9.99075
 "QX"    4.179000561482313
 "UA"    6.031844741235393
 "WN"    4.352488296605608
 "YV"    9.751735802029554
 "YX"    6.543214992862734
 ubuntu@ip-172-31-79-8:~$
```

### JAN 2020

***Best on-time performance:*** *HA, WN, DL, YX, 9E*

***Worst on-time performance:*** *OH, YV, G4, OO MQ*



```
[ubuntu@ip-172-31-79-8:~$ cat delays2020/part-00000 | more
 "9E"    9.17598855359001
 "AA"    11.035307204082969
 "AS"    11.665719293259437
 "B6"    10.356645620851546
 "DL"    7.05993655392916
 "EV"    10.888090057705043
 "F9"    9.814730694354315
 "G4"    14.551568930041153
 "HA"    4.9020611229566455
 "MQ"    13.450734873067379
 "NK"    9.667200640768923
 "OH"    15.786334279484965
 "OO"    14.328246420190267
 "UA"    9.369631236442515
 "WN"    5.32951636559081
 "YV"    15.264976441552614
 "YX"    7.959094655859321
 ubuntu@ip-172-31-79-8:~$
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## ***Mapper1.py***

```
#!/usr/bin/env python3

import sys

i = 0
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split(',')

    # increase counters
#    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        #print("{0}\t".format(word))
    if(words[14] != ""):
        arr_delay = words[14]
        airline = words[1]
    if(i > 0):
        print("{0}\t{1}".format(airline, arr_delay))
    i += 1
```

## ***Reducer1.py***

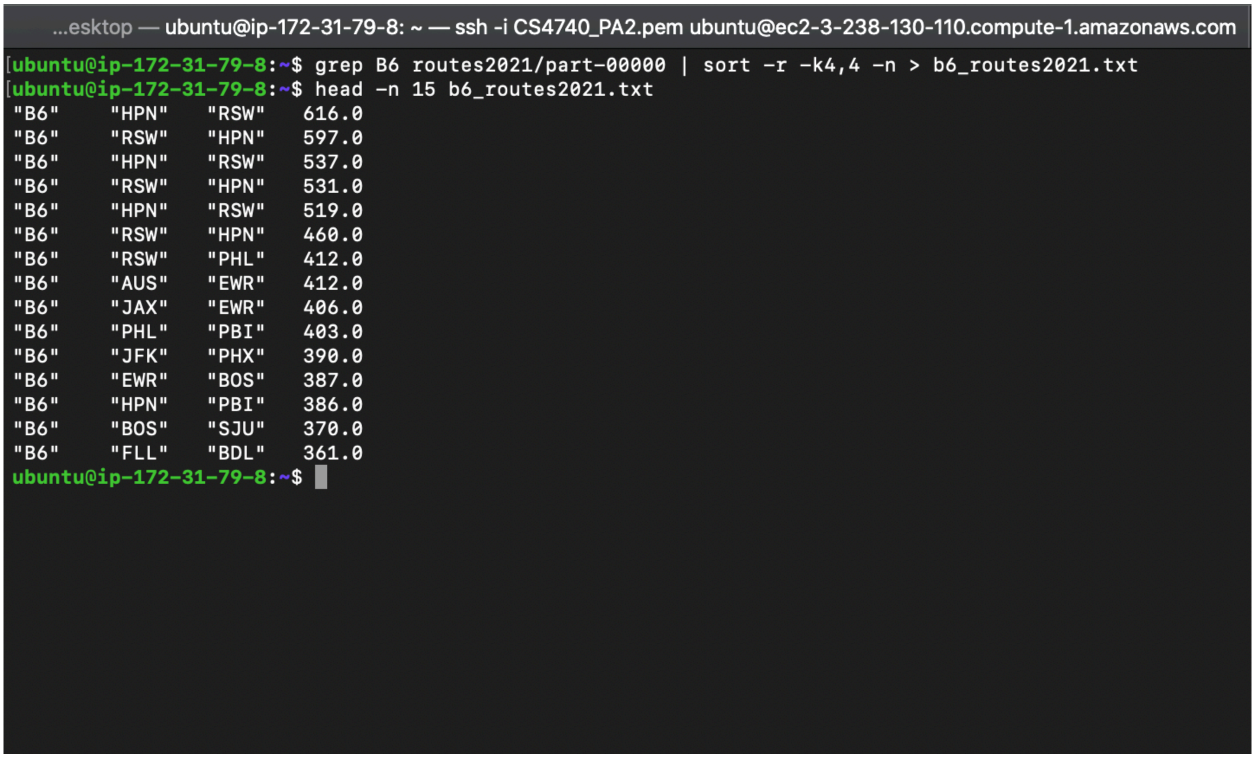
```
#!/usr/bin/env python3

from operator import itemgetter
import sys

airline_name = None
air_delay = 0
airline_count = 1
airline = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    airline, delay = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        delay = int(float(delay))
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if airline_name == airline:
        airline_count += 1
        air_delay += delay
    else:
        if airline_name:
            # write result to STDOUT
            print("{0}\t{1}".format(airline_name, float(air_delay)/airline_count))
        air_delay = delay
        airline_name = airline
        airline_count = 1

# do not forget to output the last word if needed!
if airline_name == airline:
    print("{0}\t{1}".format(airline_name, float(air_delay)/airline_count))
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## Task 2:

**JAN 2021 –** ***B6, G4, MQ***

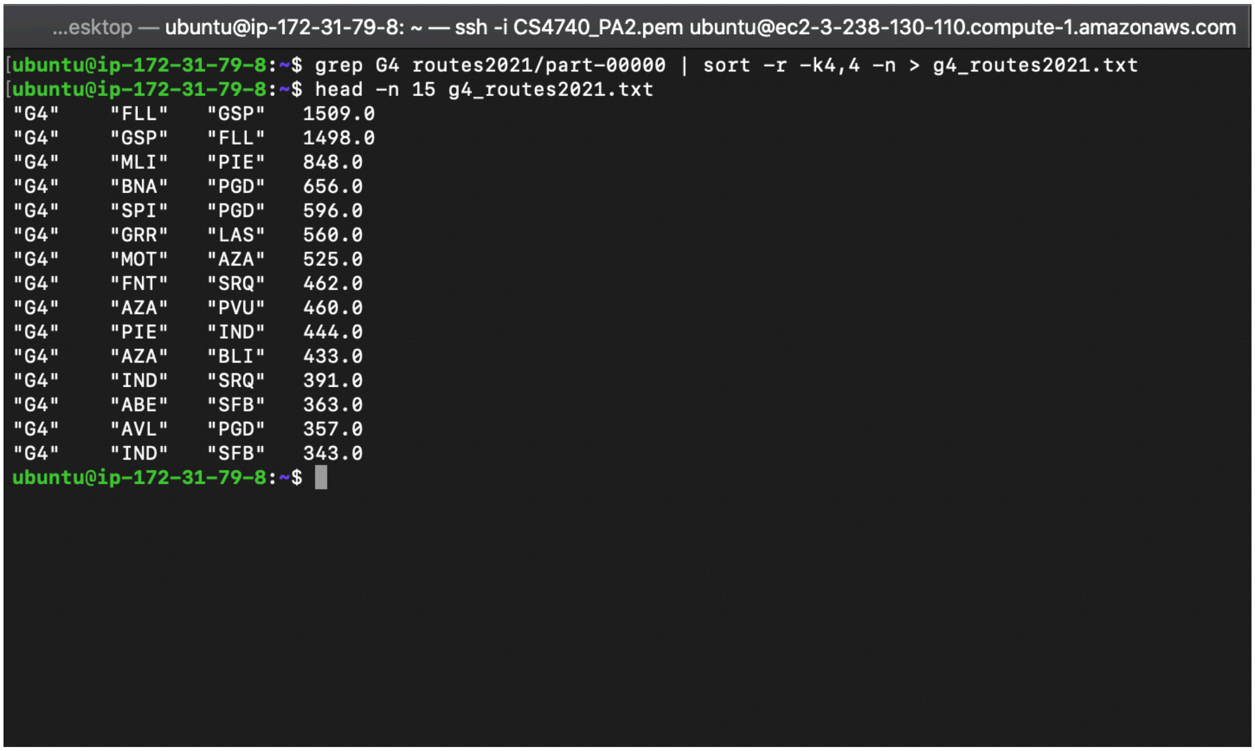
### ***B6:***



...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep B6 routes2021/part-00000 | sort -r -k4,4 -n > b6_routes2021.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 b6_routes2021.txt
"B6"    "HPN"   "RSW"   616.0
"B6"    "RSW"   "HPN"   597.0
"B6"    "HPN"   "RSW"   537.0
"B6"    "RSW"   "HPN"   531.0
"B6"    "HPN"   "RSW"   519.0
"B6"    "RSW"   "HPN"   460.0
"B6"    "RSW"   "PHL"   412.0
"B6"    "AUS"   "EWR"   412.0
"B6"    "JAX"   "EWR"   406.0
"B6"    "PHL"   "PBI"   403.0
"B6"    "JFK"   "PHX"   390.0
"B6"    "EWR"   "BOS"   387.0
"B6"    "HPN"   "PBI"   386.0
"B6"    "BOS"   "SJU"   370.0
"B6"    "FLL"   "BDL"   361.0
ubuntu@ip-172-31-79-8:~$ 
```

### ***G4:***

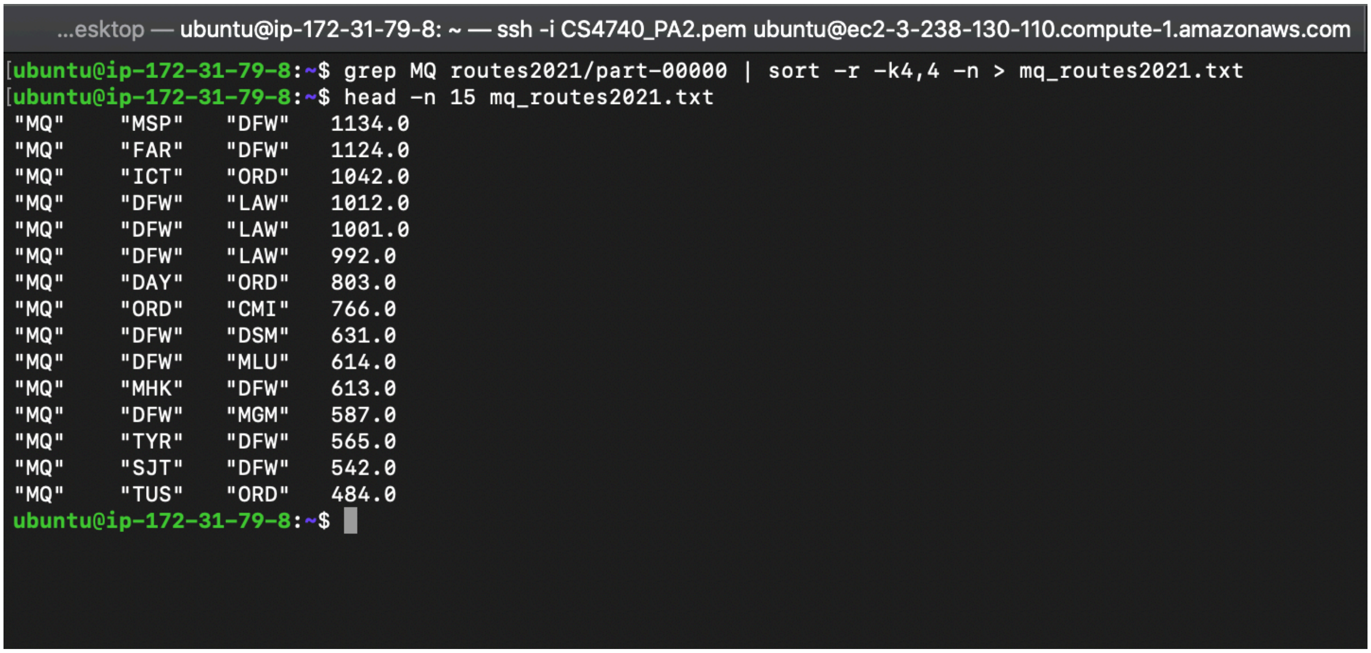


...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep G4 routes2021/part-00000 | sort -r -k4,4 -n > g4_routes2021.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 g4_routes2021.txt
"G4"    "FLL"   "GSP"   1509.0
"G4"    "GSP"   "FLL"   1498.0
"G4"    "MLI"   "PIE"   848.0
"G4"    "BNA"   "PGD"   656.0
"G4"    "SPI"   "PGD"   596.0
"G4"    "GRR"   "LAS"   560.0
"G4"    "MOT"   "AZA"   525.0
"G4"    "FNT"   "SRQ"   462.0
"G4"    "AZA"   "PVU"   460.0
"G4"    "PIE"   "IND"   444.0
"G4"    "AZA"   "BLI"   433.0
"G4"    "IND"   "SRQ"   391.0
"G4"    "ABE"   "SFB"   363.0
"G4"    "AVL"   "PGD"   357.0
"G4"    "IND"   "SFB"   343.0
ubuntu@ip-172-31-79-8:~$ 
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

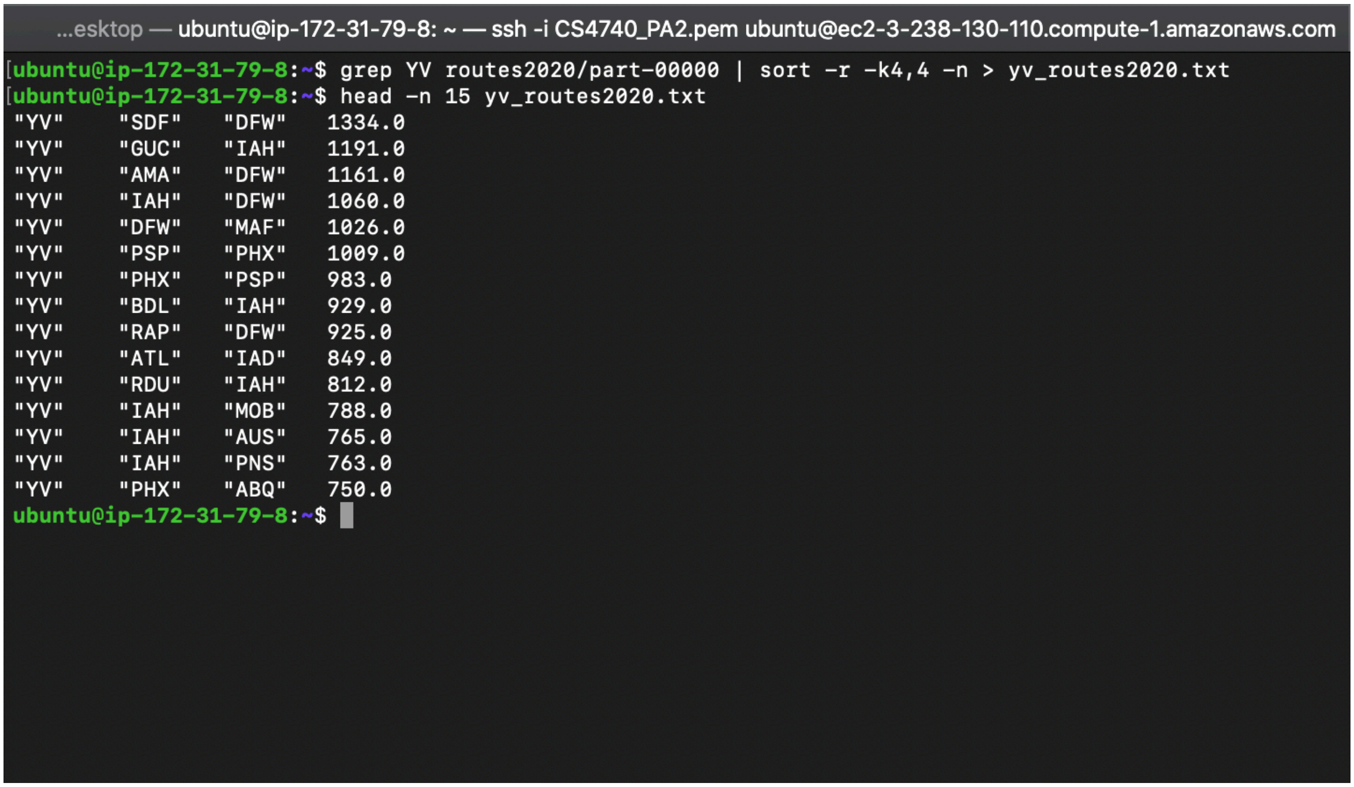
## *MQ:*


...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep MQ routes2021/part-00000 | sort -r -k4,4 -n > mq_routes2021.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 mq_routes2021.txt
 "MQ"    "MSP"   "DFW"   1134.0
 "MQ"    "FAR"   "DFW"   1124.0
 "MQ"    "ICT"   "ORD"   1042.0
 "MQ"    "DFW"   "LAW"   1012.0
 "MQ"    "DFW"   "LAW"   1001.0
 "MQ"    "DFW"   "LAW"   992.0
 "MQ"    "DAY"   "ORD"   803.0
 "MQ"    "ORD"   "CMI"   766.0
 "MQ"    "DFW"   "DSM"   631.0
 "MQ"    "DFW"   "MLU"   614.0
 "MQ"    "MHK"   "DFW"   613.0
 "MQ"    "DFW"   "MGM"   587.0
 "MQ"    "TYR"   "DFW"   565.0
 "MQ"    "SJT"   "DFW"   542.0
 "MQ"    "TUS"   "ORD"   484.0
 ubuntu@ip-172-31-79-8:~$ 
```

**JAN 2020 –** *OH, YV, G4*

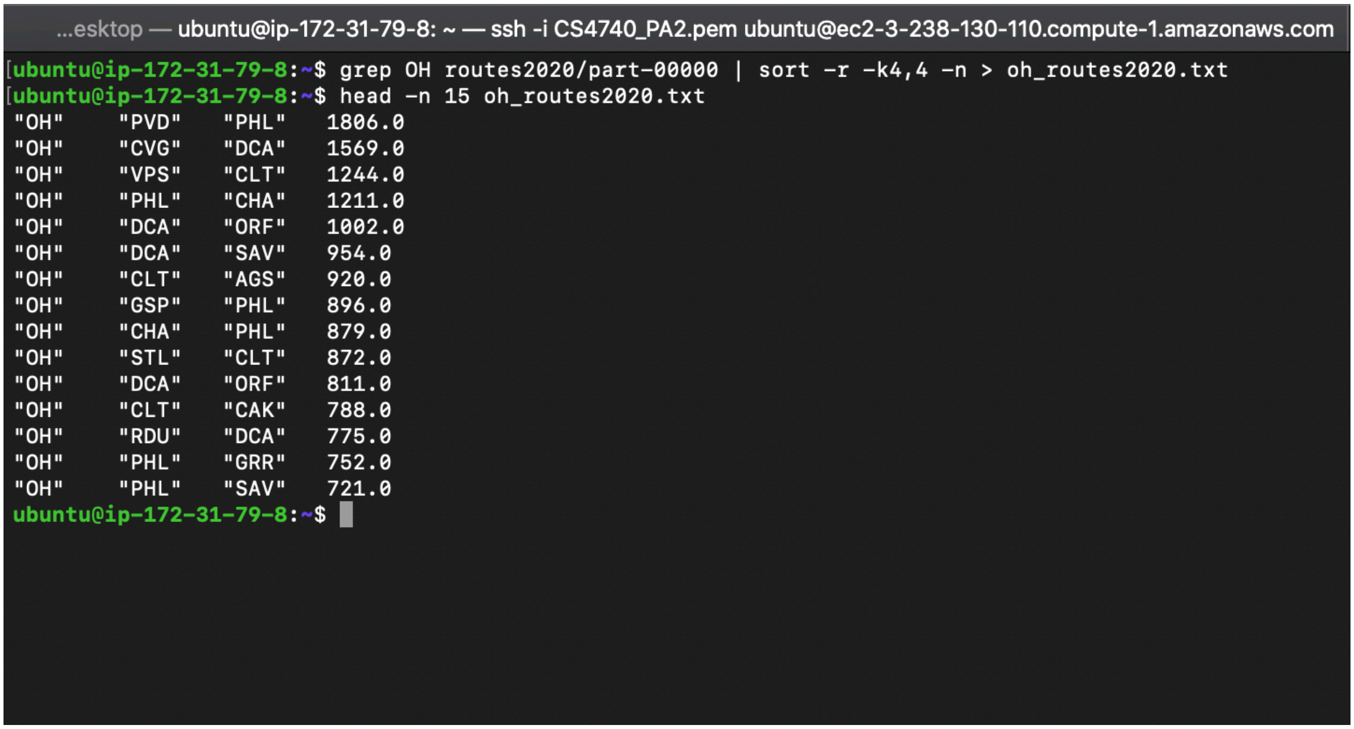
## *YV:*


...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep YV routes2020/part-00000 | sort -r -k4,4 -n > yv_routes2020.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 yv_routes2020.txt
 "YV"    "SDF"   "DFW"   1334.0
 "YV"    "GUC"   "IAH"   1191.0
 "YV"    "AMA"   "DFW"   1161.0
 "YV"    "IAH"   "DFW"   1060.0
 "YV"    "DFW"   "MAF"   1026.0
 "YV"    "PSP"   "PHX"   1009.0
 "YV"    "PHX"   "PSP"   983.0
 "YV"    "BDL"   "IAH"   929.0
 "YV"    "RAP"   "DFW"   925.0
 "YV"    "ATL"   "IAD"   849.0
 "YV"    "RDU"   "IAH"   812.0
 "YV"    "IAH"   "MOB"   788.0
 "YV"    "IAH"   "AUS"   765.0
 "YV"    "IAH"   "PNS"   763.0
 "YV"    "PHX"   "ABQ"   750.0
 ubuntu@ip-172-31-79-8:~$ 
```

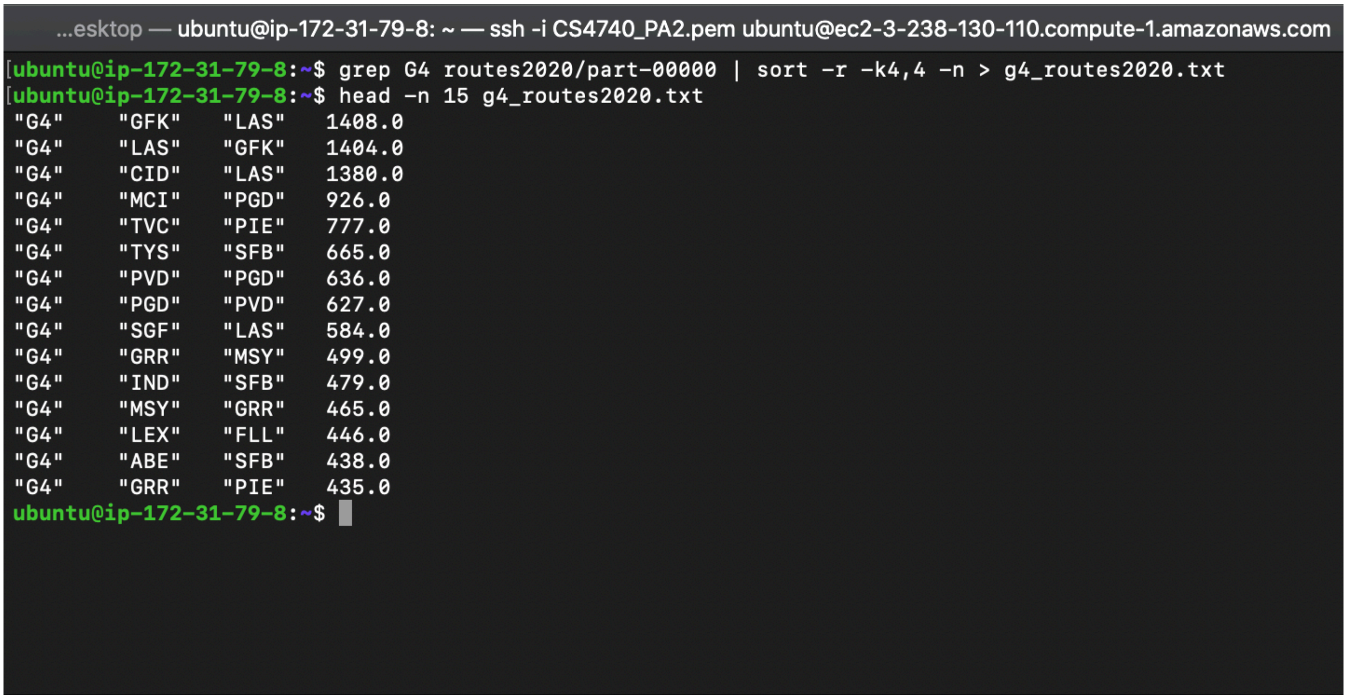
Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## ***OH:***


...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep OH routes2020/part-00000 | sort -r -k4,4 -n > oh_routes2020.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 oh_routes2020.txt
 "OH"    "PVD"   "PHL"   1806.0
 "OH"    "CVG"   "DCA"   1569.0
 "OH"    "VPS"   "CLT"   1244.0
 "OH"    "PHL"   "CHA"   1211.0
 "OH"    "DCA"   "ORF"   1002.0
 "OH"    "DCA"   "SAV"   954.0
 "OH"    "CLT"   "AGS"   920.0
 "OH"    "GSP"   "PHL"   896.0
 "OH"    "CHA"   "PHL"   879.0
 "OH"    "STL"   "CLT"   872.0
 "OH"    "DCA"   "ORF"   811.0
 "OH"    "CLT"   "CAK"   788.0
 "OH"    "RDU"   "DCA"   775.0
 "OH"    "PHL"   "GRR"   752.0
 "OH"    "PHL"   "SAV"   721.0
 ubuntu@ip-172-31-79-8:~$
```

## ***G4:***


...esktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-238-130-110.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ grep G4 routes2020/part-00000 | sort -r -k4,4 -n > g4_routes2020.txt
[ubuntu@ip-172-31-79-8:~$ head -n 15 g4_routes2020.txt
 "G4"    "GFK"   "LAS"   1408.0
 "G4"    "LAS"   "GFK"   1404.0
 "G4"    "CID"   "LAS"   1380.0
 "G4"    "MCI"   "PGD"   926.0
 "G4"    "TVC"   "PIE"   777.0
 "G4"    "TYS"   "SFB"   665.0
 "G4"    "PVD"   "PGD"   636.0
 "G4"    "PGD"   "PVD"   627.0
 "G4"    "SGF"   "LAS"   584.0
 "G4"    "GRR"   "MSY"   499.0
 "G4"    "IND"   "SFB"   479.0
 "G4"    "MSY"   "GRR"   465.0
 "G4"    "LEX"   "FLL"   446.0
 "G4"    "ABE"   "SFB"   438.0
 "G4"    "GRR"   "PIE"   435.0
 ubuntu@ip-172-31-79-8:~$
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## ***Mapper2.py***

```
#!/usr/bin/env python3

import sys

i = 0
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split(',')

    # increase counters
#    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
    if(words[14] != "" and words[4] != "" and words[9] != ""):
        arr_delay = words[14]
        origin = words[4]
        dest = words[9]
        airline = words[1]
    if(i > 0):
        if((airline == "\"B6\"" or airline == "\"G4\"" or airline == "\"MQ\"")):
            print("{0}\t{1}\t{2}\t{3}".format(airline, origin, dest, arr_delay))
    i += 1
```

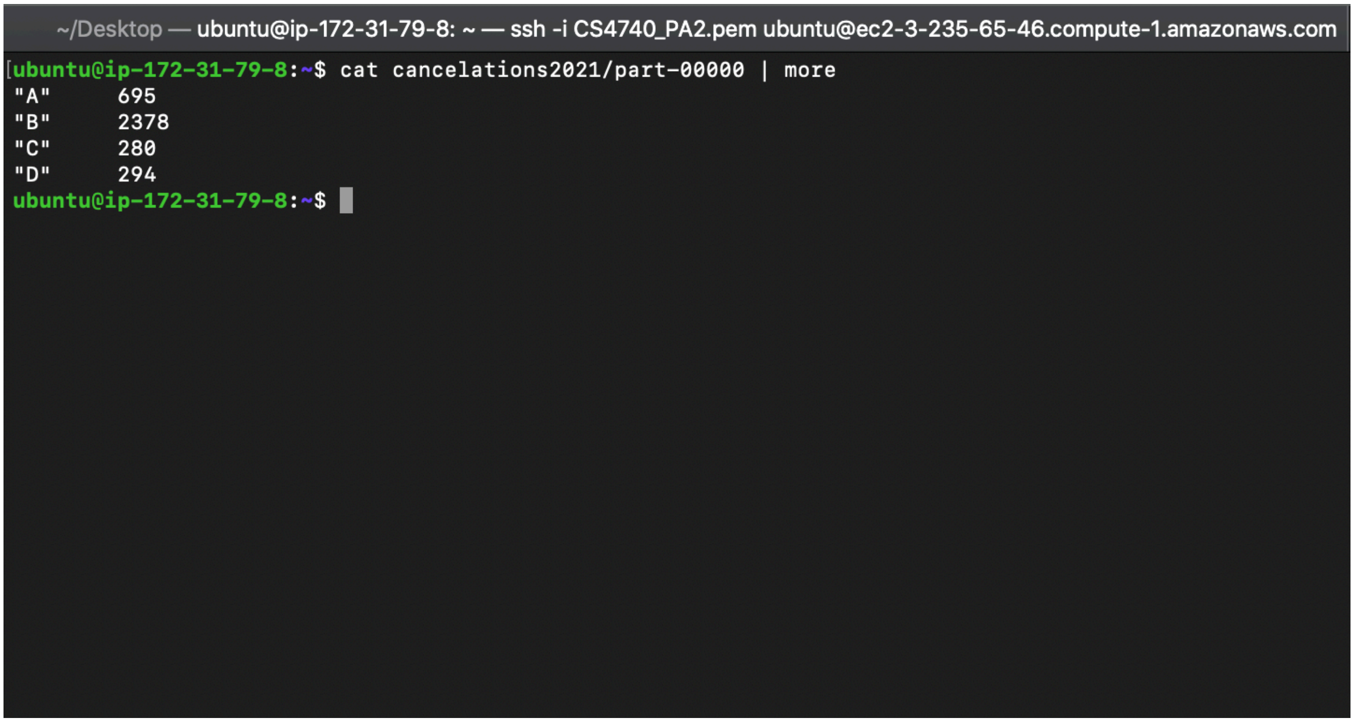
## ***Reducer2.py***

```
#!/usr/bin/env python3

from operator import itemgetter
import sys

def sortLast(val):
    return val[3]

airline_name = None
air_delay = 0
air_orig = None
air_dest = None
od_count = 1
airline_count = 1
airline = None
# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    airline, origin, dest, delay = line.split('\t', 3)

    # convert count (currently a string) to int
    try:
        delay = int(float(delay))
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if(airline_name == airline and origin == air_orig and dest == air_dest):
        od_count += 1
        air_delay += delay
    else:
        if airline_name:
            # write result to STDOUT
            print("{0}\t{1}\t{2}\t{3}".format(airline_name, air_orig, air_dest,
float(air_delay)/od_count))
        air_delay = delay
        airline_name = airline
        air_orig = origin
        air_dest = dest
        od_count = 1

# do not forget to output the last word if needed!
if airline_name == airline and origin == air_orig and dest == air_dest:
    print("{0}\t{1}\t{2}\t{3}".format(airline_name, air_orig, air_dest,
float(air_delay)/od_count))
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

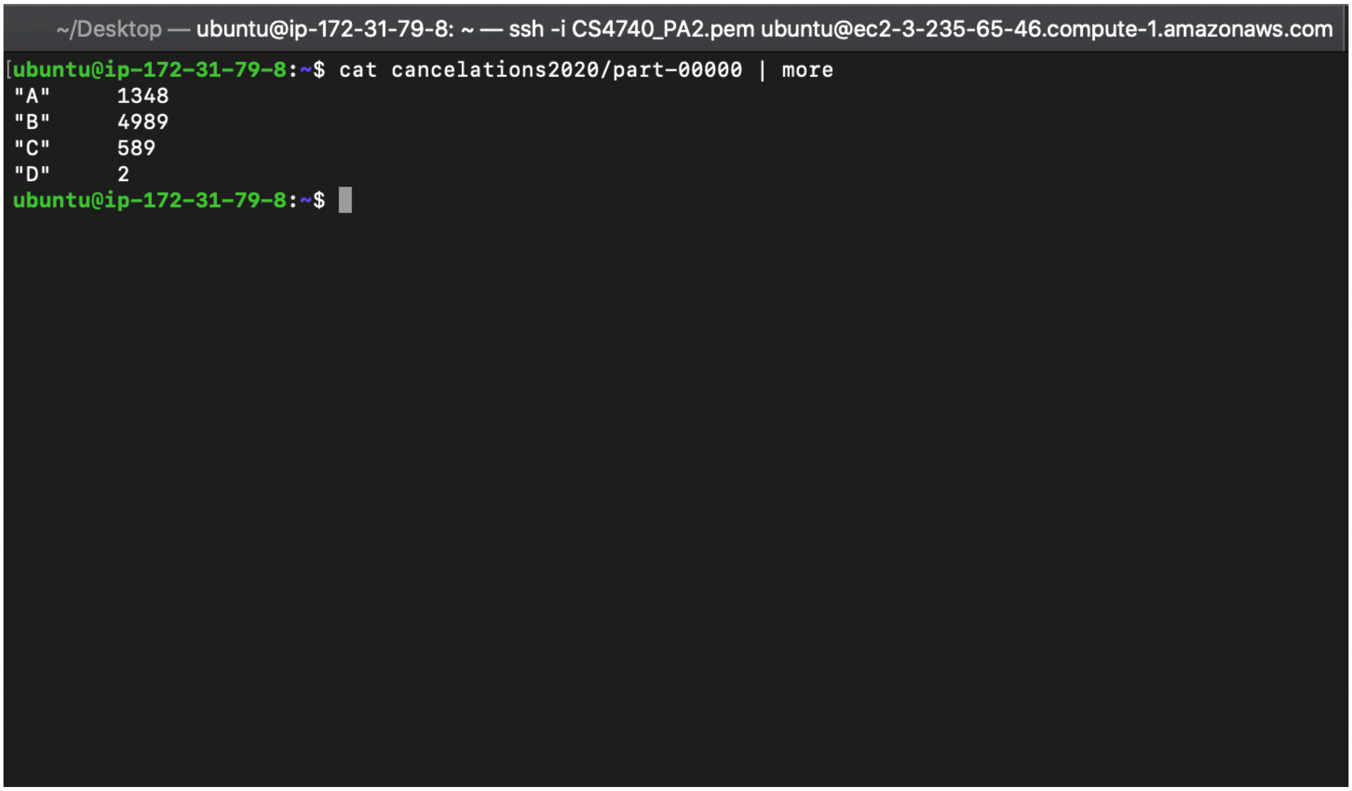
## Task 3:

### JAN 2021


~/Desktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-235-65-46.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ cat cancelations2021/part-00000 | more
 "A"     695
 "B"     2378
 "C"     280
 "D"     294
 ubuntu@ip-172-31-79-8:~$ 
```

### JAN 2020


~/Desktop — ubuntu@ip-172-31-79-8: ~ — ssh -i CS4740_PA2.pem ubuntu@ec2-3-235-65-46.compute-1.amazonaws.com

```
[ubuntu@ip-172-31-79-8:~$ cat cancelations2020/part-00000 | more
 "A"     1348
 "B"     4989
 "C"     589
 "D"     2
 ubuntu@ip-172-31-79-8:~$ 
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## ***Mapper3.py***

```
#!/usr/bin/env python3

import sys
i = 0
#airline = None
#code = None
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split(',')
    # increase counters
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
       #
        # tab-delimited; the trivial word count is 1
    if(i > 0):
        if(words[15] == "1.00" and words[16] != ""):
            print("{0}\t{1}".format(words[16], 1))
    i += 1
```

## ***reducer3.py***

```
#!/usr/bin/env python3

from operator import itemgetter
import sys

current_canc = None
current_count = 0
code = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    code, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_canc == code:
        current_count += count
    else:
        if current_canc:
            # write result to STDOUT
            print("{0}\t{1}".format(current_canc, current_count))
        current_count = count
        current_canc = code

# do not forget to output the last word if needed!
if current_canc == code:
    print("{0}\t{1}".format(current_canc, current_count))
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

## Task 4:

In 2021 and 2020, airlines HA and WN were in the top five for best on-time performance. Likewise, airlines MQ, OH, OO, and G4 were in the top five for worst on-time performance in both 2021 and 2020. For the worst routes, G4 was the only common airline for the worst on-time performance between 2020 and 2021. The top 15 worst routes differed for G4 between 2021 and 2020, where the worst route in 2021 was “FLL” to “GSP” and the worst route in 2020 was “GFK” to “LAS”. For flight cancellations, in both 2021 and 2020, flights were cancelled the most for reason “B” or weather. Additionally, the number of flights that were cancelled less for carrier and National air system reasons decreased between 2020 and 2021 while the number of flights cancelled for security increased. Overall, the number of flight cancellations increased from 2020 to 2021.